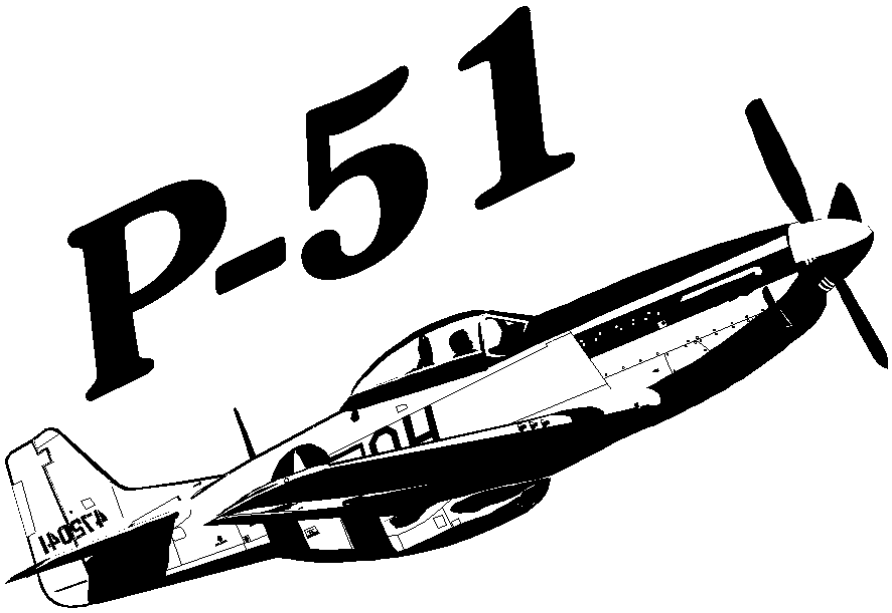


The P-51 Peripheral 8051 System on a Chip

Cybernetic Micro Systems **www.ControlChips.com**

US patent and copyright law protects Cybernetic Micro Systems' hardware and software products. This intellectual property shall remain the property of Cybernetic Micro Systems. Duplication is subject to a license from Cybernetic Micro Systems. Cybernetic Micro Systems reserves the right to improve design or performance characteristics. Cybernetic Micro Systems assumes no responsibility for the use of any circuitry other than circuitry embodied in Cybernetic Micro Systems products. No other circuit patent licenses are implied.

Information furnished by Cybernetic Micro Systems is believed to be accurate and reliable. However, no responsibility is assumed by Cybernetic Micro Systems for its use, nor for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent right of Cybernetic Micro Systems. Further Cybernetic Micro Systems reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation to notify any person or organization of such revision or changes. Cybernetic Micro Systems assumes no responsibility for any errors that may appear in this document and makes not commitment to update the information contained herein.

Because of the wide variety of applications in which this product may be used, Cybernetic Micro Systems makes no claim as to the product's fitness for any given application. It is therefore the user's responsibility to ensure the safety and viability of using this product in the user's application.

The following are trademarks and U.S. patents of Cybernetic Micro Systems, Inc.:

Trademarks				**Patents**
Bin-ASCII	CY123	CY300	CY525	5,452,301
CYMPL	CY132	CY325	CY545	5,541,930
Analog-ASCII	CY232	CY327	CY550	5,680,589
ASCII-Analog	CY233	CY308	CY600	5,721,729
CY233-LINC	CY250	CY480	CY750	5,860,021
USB-RAM	CY251	CY500		6,021,453
P-51	CY252	CY512		

Cybernetic Micro Systems, Inc.
PO Box 3000
San Gregorio CA 94074
Tel: 650-726-3000
Fax: 650-726-3003
www.ControlChips.com
info@ControlChips.com

P-51 Documentation

ERRATA / ADDENDA

Cybernetic Micro Systems

22 September 2000
29 January 2001
21 November 2002

==========

P51 User manual (5/30/2000) errata:

Page 4:

P-51 Pin List

RFRSH (pin 11 - input) description should be reversed to say:

ISA Refresh, low during a refresh cycle, high during a normal cycle.

When connected to the ISA bus, the ISA Refresh signal drives the P-51 RFRSH input directly. The pin has an internal pullup so if it is connected to a micro-controller, the P-51 refresh signal should be pulled high or left open.

Page 22:

was:

9. ...(Base + 0x3E2F => P-51 Code Address 0x2E2F)

should be:

9. ...(Base + 0x1E2F => P-51 Code Address 0x0E2F)

Page 22:

was:

11. After the Wait flag is cleared, the P-51 will execute the code from address 0x2E2F ...

should be:

11. After the Wait flag is cleared, the P-51 will execute the code from address 0x0E2F ...

Page 33, Last item in "Special Memory Function Locations" Table:

was:

BASE + 0x3E2F
Breakpoint Subroutine
Code 0x2E2F
Start of P51 Subroutine executed after a breakpoint

should be:

BASE + 0x1E2F
Breakpoint Subroutine
Code 0x0E2F
Start of P51 Subroutine executed after a breakpoint

==========

Addenda:

Movx Latch

The "movx" instructions in the P51 are identical to the "movx" instructions in the 8051, with respect to the read and write signals, P3.6 and P3.7, and to address and data on P2 and P0. This is true whether the target of the "movx" is internal or external; that is, whether the P51 is accessing the internal dual port RAM or an external RAM/IO-device.

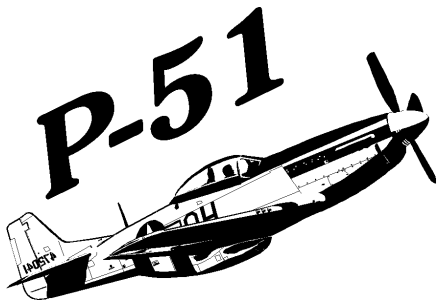
A consequence of this is that systems that use only internal dual port RAM will still see P2:P0 changes (and P3.6,P3.7) when "movx" occurs. If these ports are intended to drive external devices, then latches may be required on P2 and P0 to shield the external devices from "movx" induced changes.

Similarly, if external RAM-I/O is used, the external RAM must be disabled during internal "movx" operations. This will normally be done using P1.6 as chip select.

2E2F breakpoint address:

When a breakpoint occurs, the P51 indicates that the next instruction will be at 2E2F. Since the current version P51 has only 8K of RAM, 2E2F is outside of the address space, and execution actually occurs at 0E2F, ignoring the highest bit. Note that if you actually use 2E2F the P51 will work, ignoring the msbit and using 0E2Fh.

Table of Contents



The P-51 Peripheral 8051 System on a Chip

For two decades the 8051 microprocessor/computer has served as *the* standalone solution to countless embedded system applications. The flexibility, power, and familiarity of the 8051 are such that 8051 knowledge is in almost every designer's tool kit. A major weakness of the '51 is its interface to a host processor, typically either UART-based or parallel handshake. Because these typically mismatch the speed of the host, the Cybernetics P-51 provides a novel tightly coupled architecture for adding the power of the 8051 to a host processor.

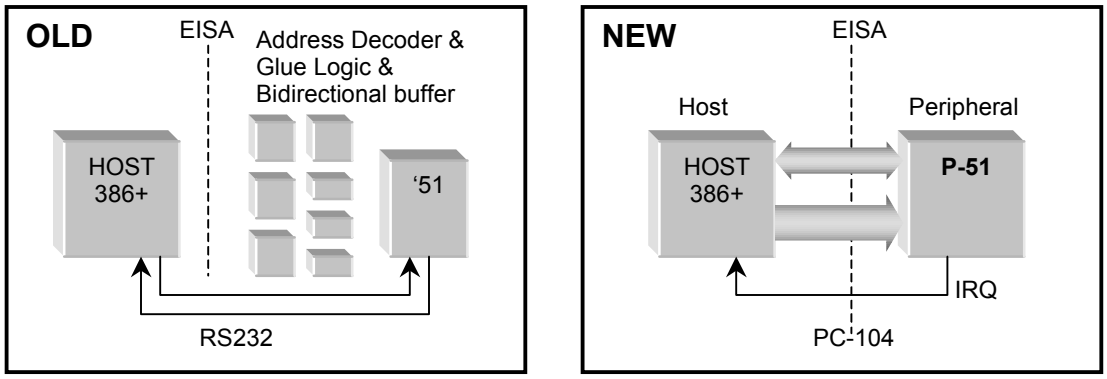


Figure 1. The P-51 peripheral 8051 interfaces **directly** to an ISA-bus, PC-104, or equivalent, instead of requiring 10 to 20 glue chips, as does an 8051 processor.

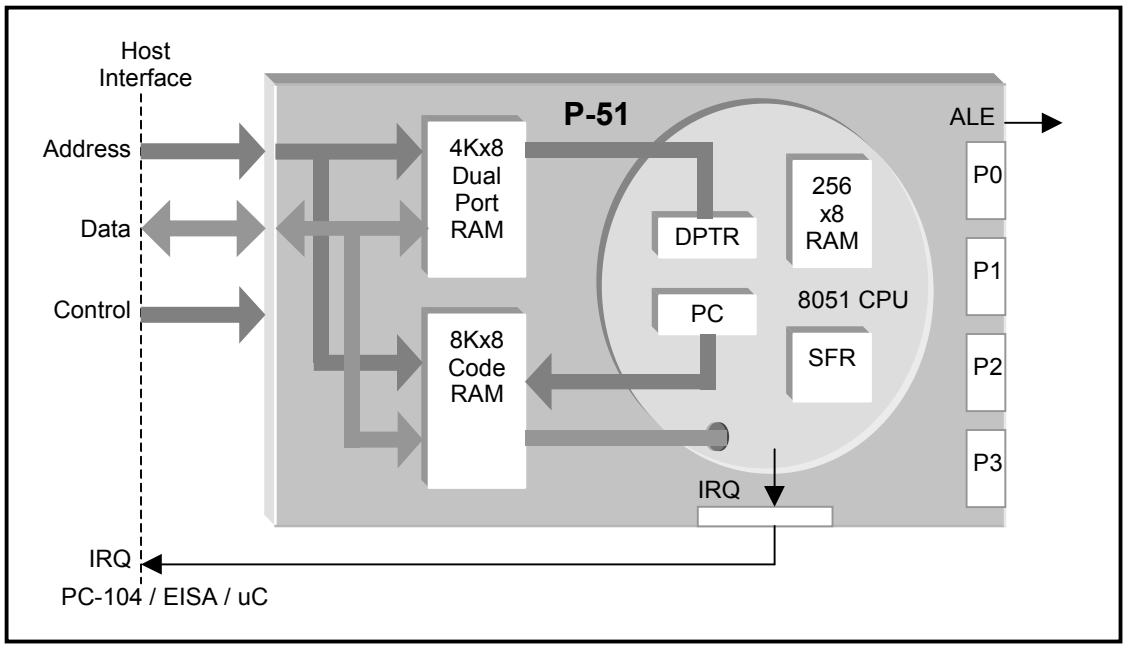
The P-51 Peripheral-8051 Architecture

Why P-51? First the name. As most engineers know, the P-51 was the fastest propeller driven fighter in WW-II, achieving 451 MPH in level flight, and exhibiting a graceful beauty and a deadly effectiveness. In honor of this marvelous design we have chosen the P-51 logo:

The P-51 name is a reminder that this device is a **Peripheral 8051 processor, <u>not</u> a standalone 8051 processor**. Most engineers have a clear image of the 8051 CPU, so *the name P-51 focuses on the unique aspect of the P-51*:

A peripheral 8051 requires a host to which it is peripheral.

This important difference is key to understanding the P-51 architecture, below:



The P-51 Host Bus Interface

The P-51 is a **peripheral** processor, and thus serves as a peripheral to a host processor. Because most of the ports, instruction set, I/O hardware, etc., are identical to the 8052 micro controller, most engineers need only understand the new host bus interface to begin using the device.

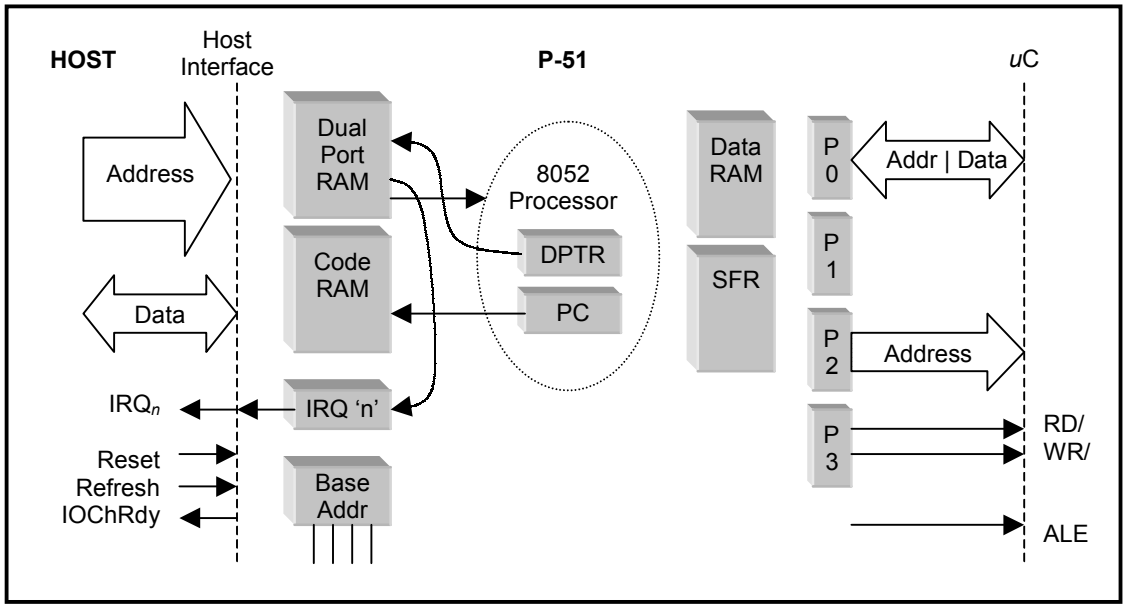
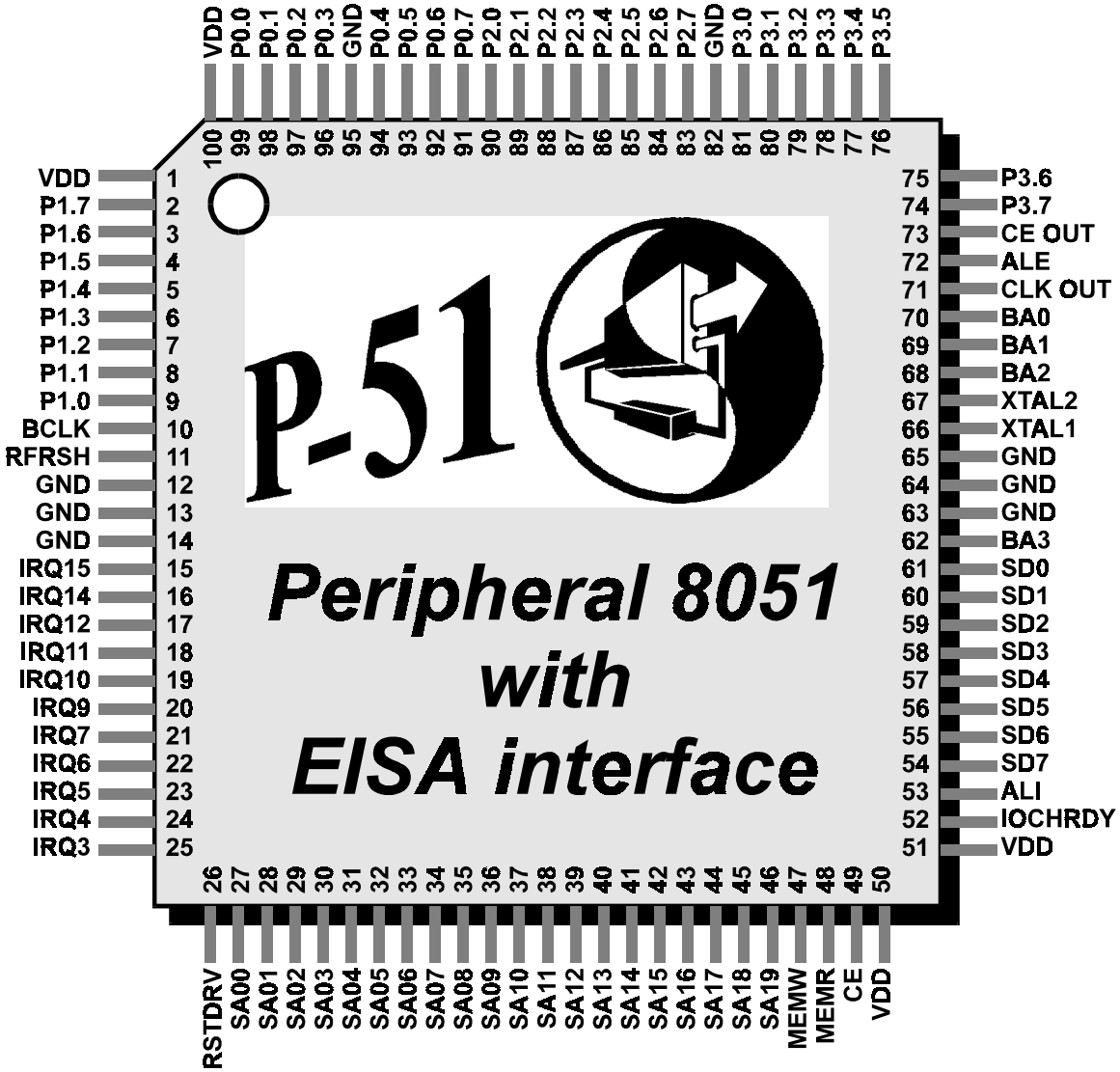


Figure 2. The host interface provides host access to Dual Port RAM and Code RAM and allows the P-51 to interrupt the host using IRQs.

As shown in Figure 2, the host processor can access the P-51's **Code RAM** and **Dual Port RAM** via a simple ISA bus compatible interface. The host provides a 20-bit address and 8-bit bi-directional data bus to the P-51. The P-51 supports the EISA **REFRESH** & **IOCHRDY** and provides a user selectable **IRQ** output.

P-51 Pinout Diagram



P-51 Pin List

Pin Name	Pin #	I/O	Description
Vdd	1, 50, 51, 100	P	3.3 Volt Power Supply
Gnd	12, 13, 14, 63, 64, 65, 82, 95	P	Ground
P0.0 - P0.7	99, 98, 97, 96, 94, 93, 92, 91	I/O	8051 Port 0 (Multiplexed Data Bus and Lower Address Byte)
P1.0 - P1.7	9, 8, 7, 6, 5, 4, 3, 2	I/O	8051 Port 1 (General I/O)
P2.0 - P2.7	90, 89, 88, 87, 86, 85, 84, 83	I/O	8051 Port 2 (General I/O and Upper Address Byte)
P3.0 - P3.7	81, 80, 79, 78, 77, 76, 75, 74	I/O	8051 Port 3 (General I/O and Special Controls)
BClk	10	I	ISA Bus Clock, normally 8.33 MHz
RFrsh	11	I	ISA Refresh, low during a refresh cycle, high during a normal cycle
IRQ3 - IRQ15	25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15	O	ISA Interrupt Request signals, open drain. No IRQ8 or IRQ13
RstDrv	26	I	ISA Reset Drive, high during system reset
SA0 - SA19	27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46	I	ISA Address Bus
MemW	47	I	ISA Memory Write strobe
MemR	48	I	ISA Memory Read strobe
CE	49	I	Chip Enable, low during ISA Read/Write cycles, may be tied low
IOChRdy	52	O	ISA I/O Channel Ready, open drain, low to extend an ISA cycle
ALI	53	I	Address Latch Input, high if SA0 to SA7 are valid as address signals, low to latch SA0 to SA7 for multiplexed Address/Data bus (non-ISA)
SD0 - SD7	61, 60, 59, 58, 57, 56, 55, 54	I/O	ISA Data Bus
BA0 - BA3	70, 69, 68, 62	I	Base Address jumper inputs
XTAL1	66	I	Crystal Oscillator circuit input, also external oscillator module input
XTAL2	67	I	Crystal Oscillator circuit input, no connect for oscillator module
Clk Out	71	O	Clock Out, P-51 oscillator circuit output clock
ALE	72	O	8051 Address Latch Enable output
CE Out	73	O	Chip Enable Output, when low, all movx instructions access external memory or devices (inverse of P1.6 signal)

P-51 Code RAM

Because its operation is simplest, we discuss the P-51 **Code RAM** first.

As a peripheral device, the P-51 expects to be reset by the host, via a signal such as the (E)ISA **RESETDRV** signal, which resets all devices at startup. While the P-51 can be placed in reset via this hardware strobe, the P-51 will remain in the reset state until released via a software operation, as described below. During the period *while the P-51 is held in* RESET:

the host has access to the P-51 **Code RAM.**

This is *a key fact of the P-51 operation*, so we repeat it.

While the P-51 is RESET, the host processor can write into the P-51's **Code RAM**. When released from RESET, *the P-51 executes from location 0 in Code RAM*!

This is the most significant difference between a "standalone" 8051/52 and the "peripheral" P-51. Standalone devices store code in non-volatile memory such as EPROM, ROM, FLASH, etc. Peripheral devices store code in volatile memory, therefore a host, or an Internet-connected device, can supply code to the P-51.

The host can access the P-51's code memory only while the P-51 is RESET.

When the P-51 executes from **Code RAM**, the host is locked out of **Code RAM**.

(The host can reset the P-51 at any time and regain access to **Code RAM**.)

Because most users are familiar with the 8051, and understand its "standalone" capability (and limitation) we emphasize (still again) the differences in operation:

The host downloads code to the P-51's **Code RAM** while the P-51 is RESET. When the P-51 is released from reset, it "*wakes up*" and immediately begins executing the downloaded code from location zero.

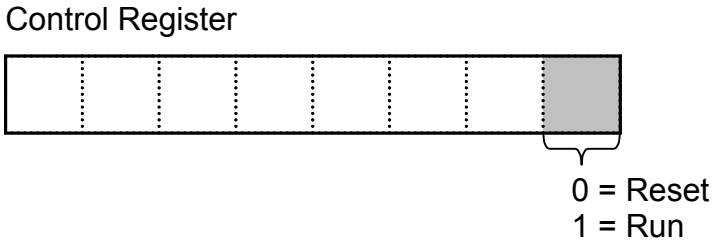
The Reset Control Register of the P-51

The P-51 possesses a host accessible control register that is mapped into dual-port RAM address space as described below. This control register allows the host to bring the P-51 in and out of reset via software, as well as driving the P-51 into the reset state via the reset pin, RSTDRV, on the P-51.

P-51 Reset Behavior

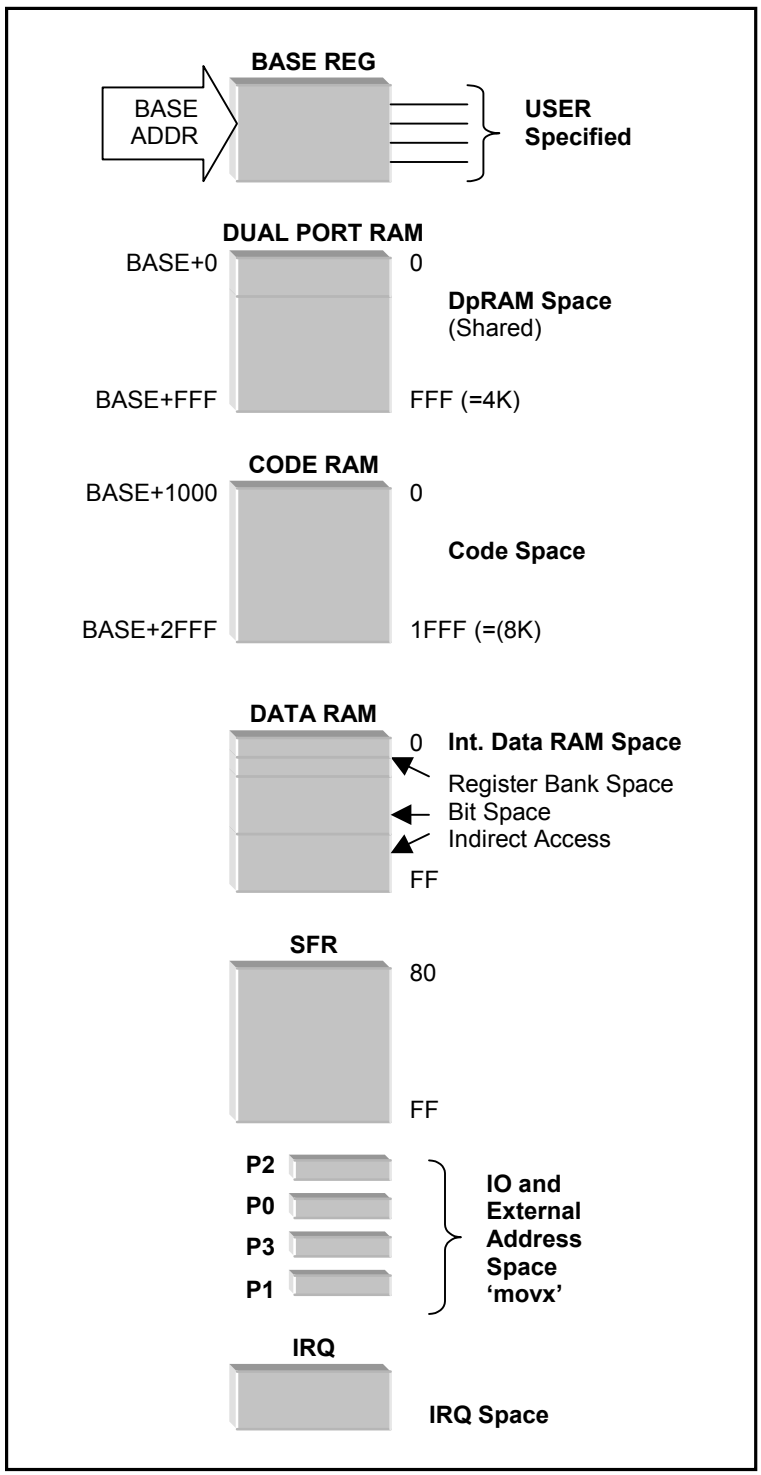
RESET-pin: The P-51 is reset when the reset pin, RSTDRV, is driven hi, and remains in reset when the RSTDRV pin returns low.

RESET-control Register: The P-51 is reset when the least significant bit of the control register is cleared to '0' and released from reset when this bit is set to '1'.



The multiple address spaces probably represent the most complex aspect of the 8051, and the P-51 is slightly more complex. The basic 8051 address spaces are relative only to the 8051, while a subset of the P-51 address spaces are also relative to the host. Although each address space effectively starts at location/ address zero from the 8051 perspective, the address as seen by the host is more complicated. The complication comes from a *base address register* whose contents are pin selectable (and software selectable) that is, the base address can be specified via four pins, connected through jumpers to ground or VCC. The high bits of the host address bus are compared to these pins. If they match, the P-51 is chip selected by the host, and its relevant address spaces become accessible to the host. Thus the P-51 can be mapped into the address space of an IBM-PC class computer and can function as truly "intelligent memory".

P-51 Address Space



As shown in this figure:

Dual port RAM is at **BASE + 0.**

Code RAM is at **BASE + 0x1000** (= **BASE + 4096**).

For example, if **BASE** is set to 0x0D (= 1101) then the base address is set to 0xE4 = 11100100 which corresponds to segment 0xE400 in an 80x86 (Real mode) segment register. The net effect is to set the ***six*** most significant bits from the appropriate base address value in the Base Table as the active Base Address. These six bits are compared to the six most significant bits of the 20-bit address applied by the host to the P-51. When these six address bits match, the P-51 is selected, else the P-51 is de-selected, and no access to P-51 Address space is available to the host.

Figure 3. The basic address spaces of the P-51.

Base Address Switch Settings:

	Switch Settings	Segment Address
All Closed:	0000	00
	0001	A4
	0010	A8
	0011	AC
	0100	B0
	0101	B4
	0110	C8
	0111	CC
	1000	D0
	1001	D4
	1010	D8
	1011	DC
	1100	E0
	1101	E4
	1110	E8
All Open	1111	EC

We consider three example cases in detail:

Jumper	Base Address	Host Address to select P-51 Base
0000	000000	0000 0000 0000 0000 0000
1010	110110	1101 1000 0000 0000 0000
0101	101101	1011 0100 0000 0000 0000

The first case is the "default base" in which both the P-51 and the host addresses begin at zero. While this may be appropriate for many embedded applications, it is typically inappropriate for IBM-PC type applications, in which address zero for the PC is dedicated to other functions.

In the case in which the Base address is set to '0101', the six base address bits are '101100' which corresponds to a host address: 1011 0100 0000 0000 0000.
When the host addresses 0xB4000 the Base address of the P-51 is selected.

Code RAM Access:

Assume that the P-51 is attached to an (E)ISA-bus and that it's Base address jumpers are set to '1010'. As can be read from the Base Address Table above, this selects a Base address of 0xD8000 for the P-51 as seen by the host. Since **Code RAM** is at Base + 0x1000, to access the first byte of **Code RAM**, the host must put address D9000 = D8000 + 1000 = 1101_1001_0000_0000_0000.

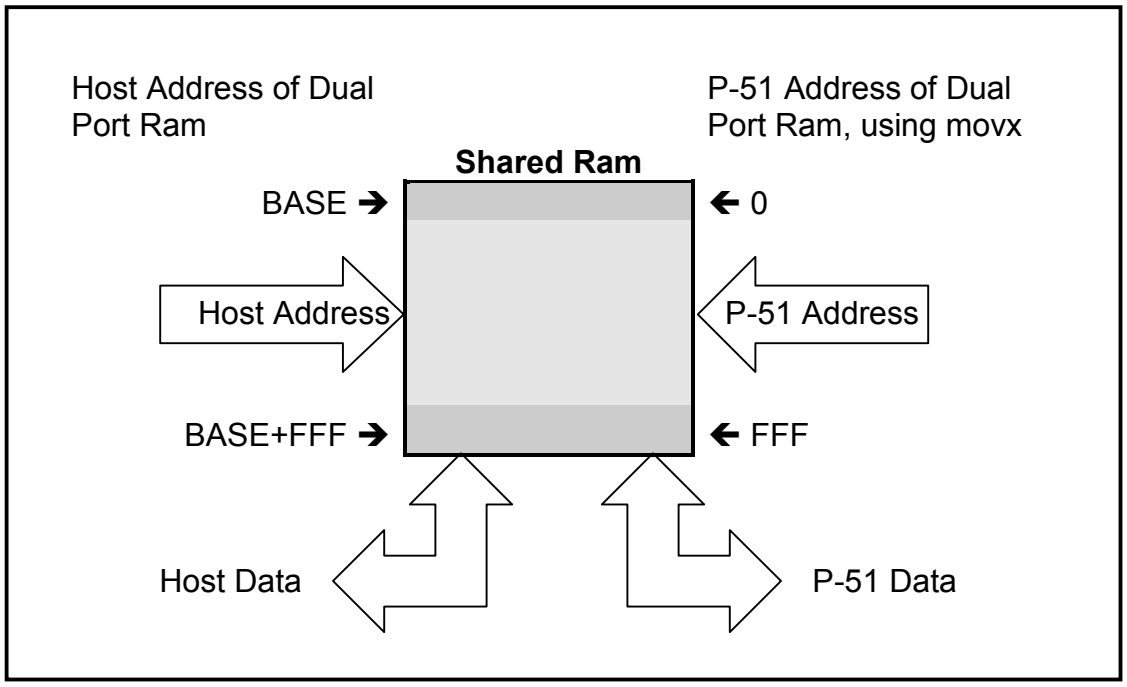
To access the second byte of Code RAM the host must put (hex) address 0xD9001 = 1101_1001_0000_0000_0001 on the 20 bit P-51 host address bus.

Using these addresses, the host can write 8051 binary opcodes into **Code RAM** that will be executed by the P-51 when the P-51 is released from **RESET**. From the P-51's perspective, the first byte of **Code RAM** is at location/address zero.

Host Access to the Dual Port RAM

As discussed above, address zero in the P-51 **Code RAM** is offset (hex) into host address space by BASE + 0x1000. Thus Code read from location 0x100, as seen by the P-51 program counter, will be written into BASE + 0x1000 + 0x100, as seen by the host.

If this is understood, then access to the **Dual Port RAM** shared memory space is simple. The first byte of **dual port RAM** is at the BASE address as seen by the host and at address zero of Data Memory as seen by the P-51:



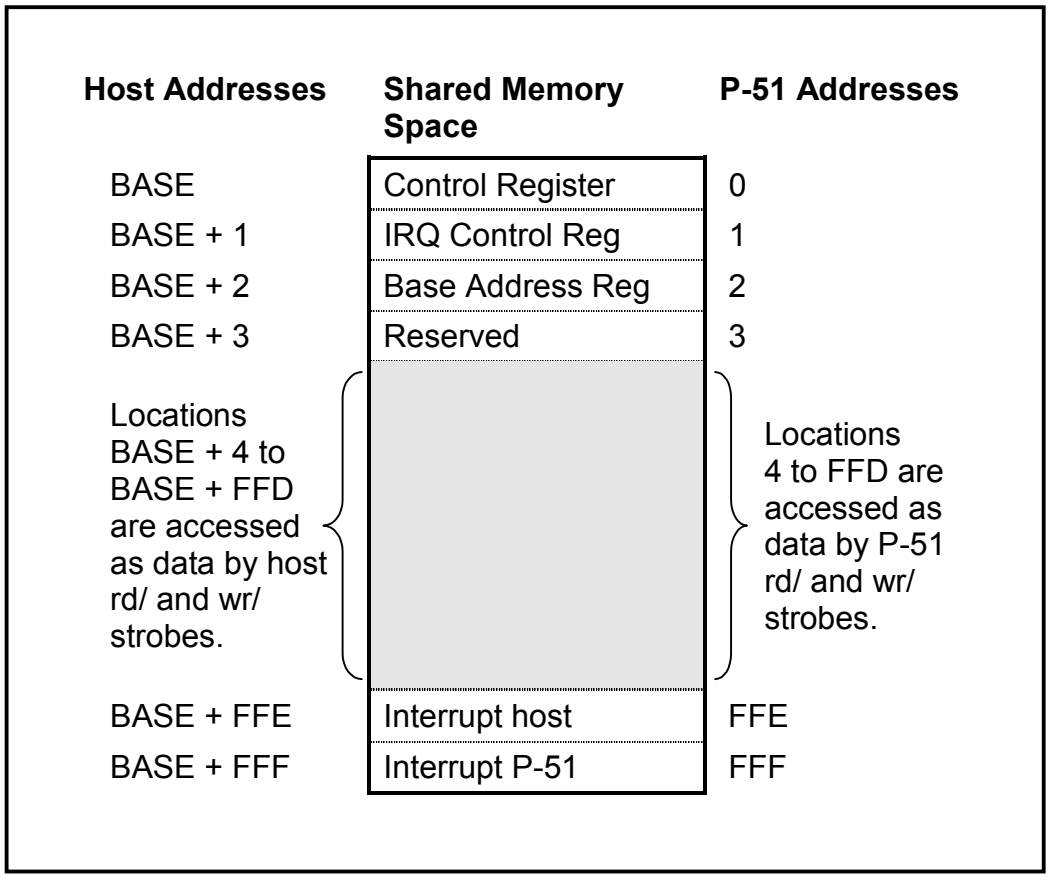
Note that, unlike **Code RAM**, which is accessible only when the P-51 is **RESET** and the BASE address + 1000 is presented to the P-51, the **dual port RAM** is accessible by the host whenever the BASE address is presented to the P-51.

Chip Enable and Read/Write strobes:

Although the host must present an address that matches the BASE address set into the BASE address latch, the P-51 does not actually allow access to **dual port RAM** (or **Code RAM**) unless the Chip Enable pin (#49) is asserted low and a read or write strobe is presented to the P-51.

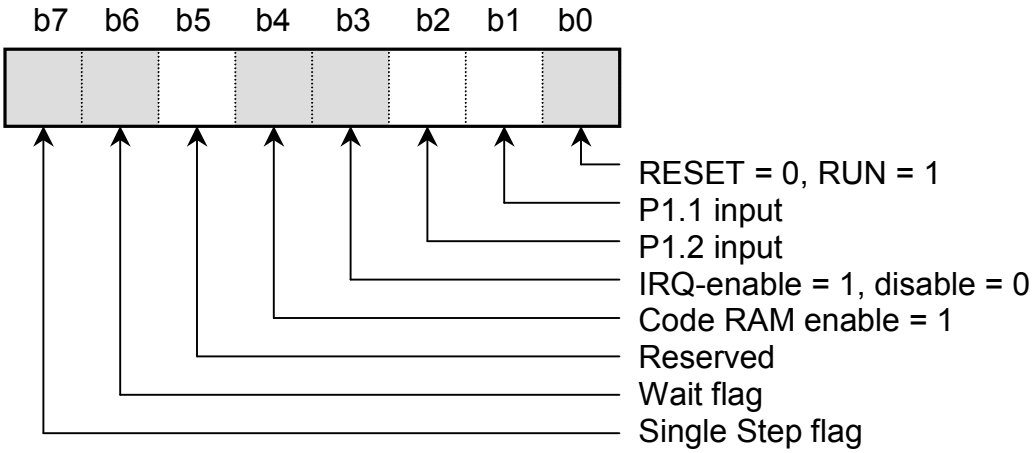
Special Control Registers in Shared Memory Space:

Now that we know how to access shared memory space from the host, we can understand how to access several control registers *mapped into* **dual port RAM** space. Specifically, the first four addresses in shared memory space access control registers, as well as **dual port RAM**. Thus, when the host writes to address BASE + 0, BASE + 1, BASE + 2 or BASE + 3, the control registers are addressed (and data is also written into **dual port RAM**). Similarly, the last two addresses in **dual port RAM** space are mapped by special registers:



The control registers at BASE to BASE + 3 are write-only, and can only be written by the host. If the P-51 writes to these locations, it only writes into the dual port RAM, not into the control registers. A read of these location returns the last data written into the dual port RAM.

Control Register Detail



The P-51 Control Register is the "master" control that causes the P-51 to be held in the RESET state or to **RUN** programs from **Code RAM**. If zero is written to this register the P-51 is reset and remains reset while CR.0 = 0. If access **to Code RAM** is desired, then CR.4 must be set to 1. Although CR.4 may be zero for manufacturing test of the P-51, all user operations, either downloading code while reset or executing code while running, require CR.4 = 1.

CR.3 is an **IRQ-enable** bit that can disable interrupts from the P-51 to the host. For example, if the host is not prepared to service interrupts from the P-51 then CR.3 should be set to zero. To allow P-51 interrupts set bit CR.3 to one.

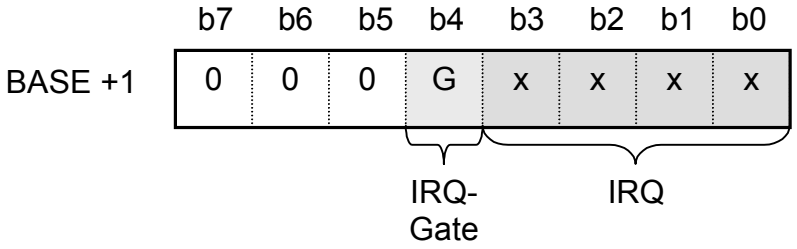
CR.1 and CR.2 control register bits are mapped into Port 1 bit 1 and Port 1 bit 2, that is, a P-51 can read CR.1 on P1.1 and CR.2 on P1.2. These user definable bits allow for example, the user to specify two "mode bits" that the P-51 software can test while executing. There are no preferred values for these bits as their value and their meanings are completely user defined. Debug Bits CR.5, CR.6 and CR.7 are defined in a later section, and should usually be set to zero.

Note that the Control Register does not allow individual bit access. That is, all eight bits must be written at once. Therefore, the values written to the control register must set or reset the desired bit while **preserving** the remaining bits.

For example, to reset the P-51 and then download code to address N of the **Code RAM** (at host address BASE + 0x1000 + N) the user/host should write 0001 0000 (0x10) to the Control Register which is located at BASE address. After the download is finished, the user can write 0001 0001 (0x11) to release the P-51 from **RESET** (but with interrupts inhibited) or write 0001 1001 (0x19) to release the P-51 to execute and to enable the IRQ output to the host.

The IRQ Control Register:

As may be seen from the pinout diagram -- the P-51 supports numerous IRQ output pins (**IRQ-3** to **IRQ-15**) covering most of the IBM-PC IRQ pins of the EISA bus. These are open collector outputs that are typically pulled up to 5 volts but can be driven low to interrupt the host. Because IRQs are a scarce resource, only one is used by the P-51 and is selected by the host via the IRQ Control Register, mapped into location BASE +1 of the host address space.

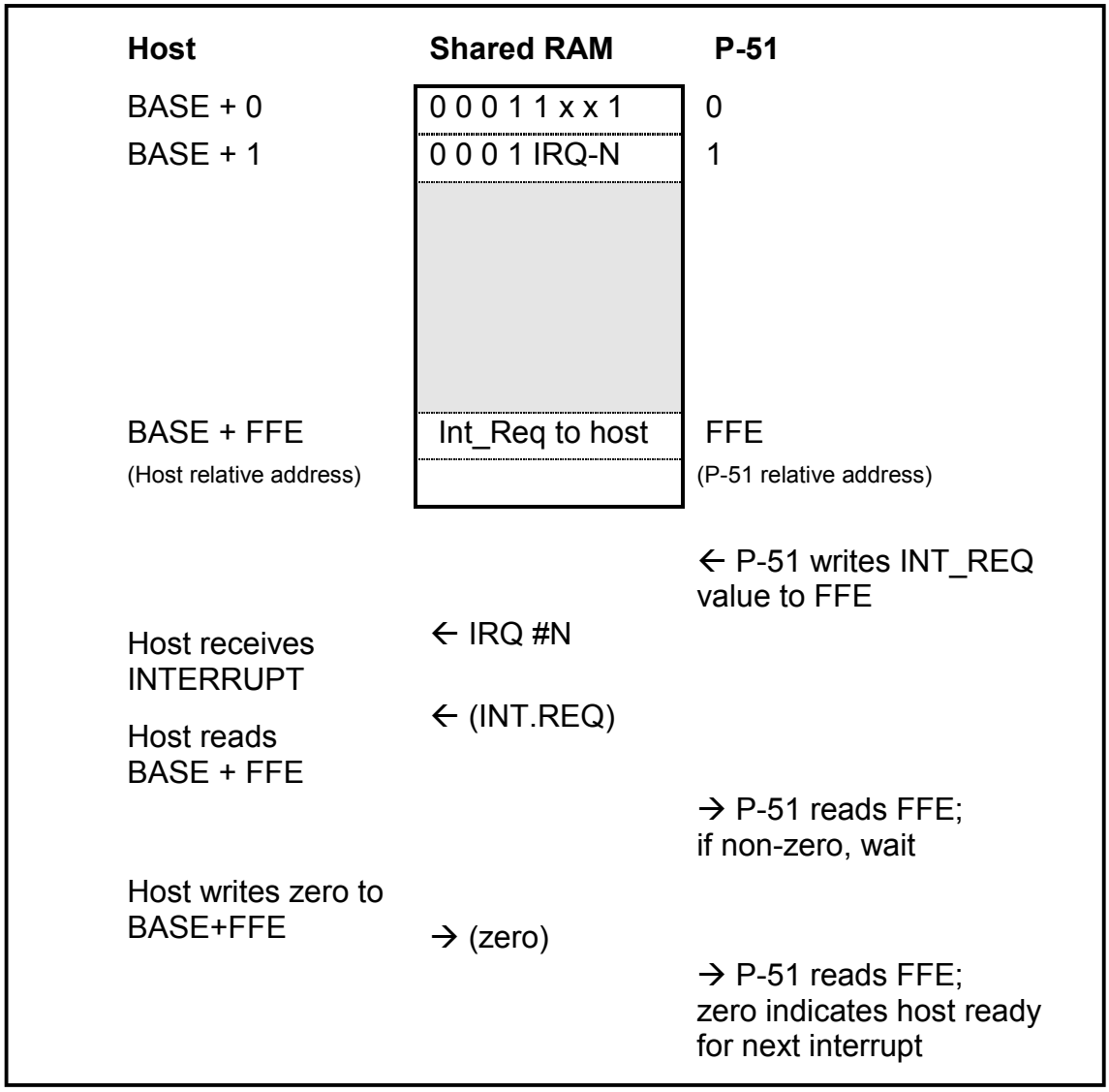


The low nibble is set to the desired IRQ#, for example, 0101 selects **IRQ-5** while 1111 selects **IRQ-15**, etc. Bit G (= IRQ.4) is the Interrupt GATE which enables or disables the IRQ (in combination with Control Register bit CR.3). The most significant bits of the IRQ Control Register must be set to zero.

Shared Memory Interrupt Locations:

Although the IRQ control gate determines which IRQ output will be used to interrupt the host, the actual interrupt mechanism is associated with the **INTERRUPT Register** at address **BASE + 0xFFE** (host perspective) or equivalent address **0xFFE** (P-51 perspective). When the P-51 writes to address **0xFFE** in shared memory space, an interrupt is generated and sent to the host on IRQ #N if the IRQ control register contains 0001 N and the Control Register = 0001 1xx1. The host is assumed to contain an appropriate ISR (Interrupt Service Routine) which will read the interrupt register at (**BASE + 0xFFE**) and will interpret the value read from this location. The interrupt appearing on IRQ #N is a low going pulse. When this pulse has been generated (by the P-51 writing to **0xFFE**), all further interrupt pulses to the host are inhibited until the host reads the **(BASE +) 0xFFE** location. After the host reads this location another interrupt can be generated by the P-51, since the "lock" is removed by the read. However a P-51 program does not "know" that the interrupt has been cleared, therefore the appropriate protocol is as follows:

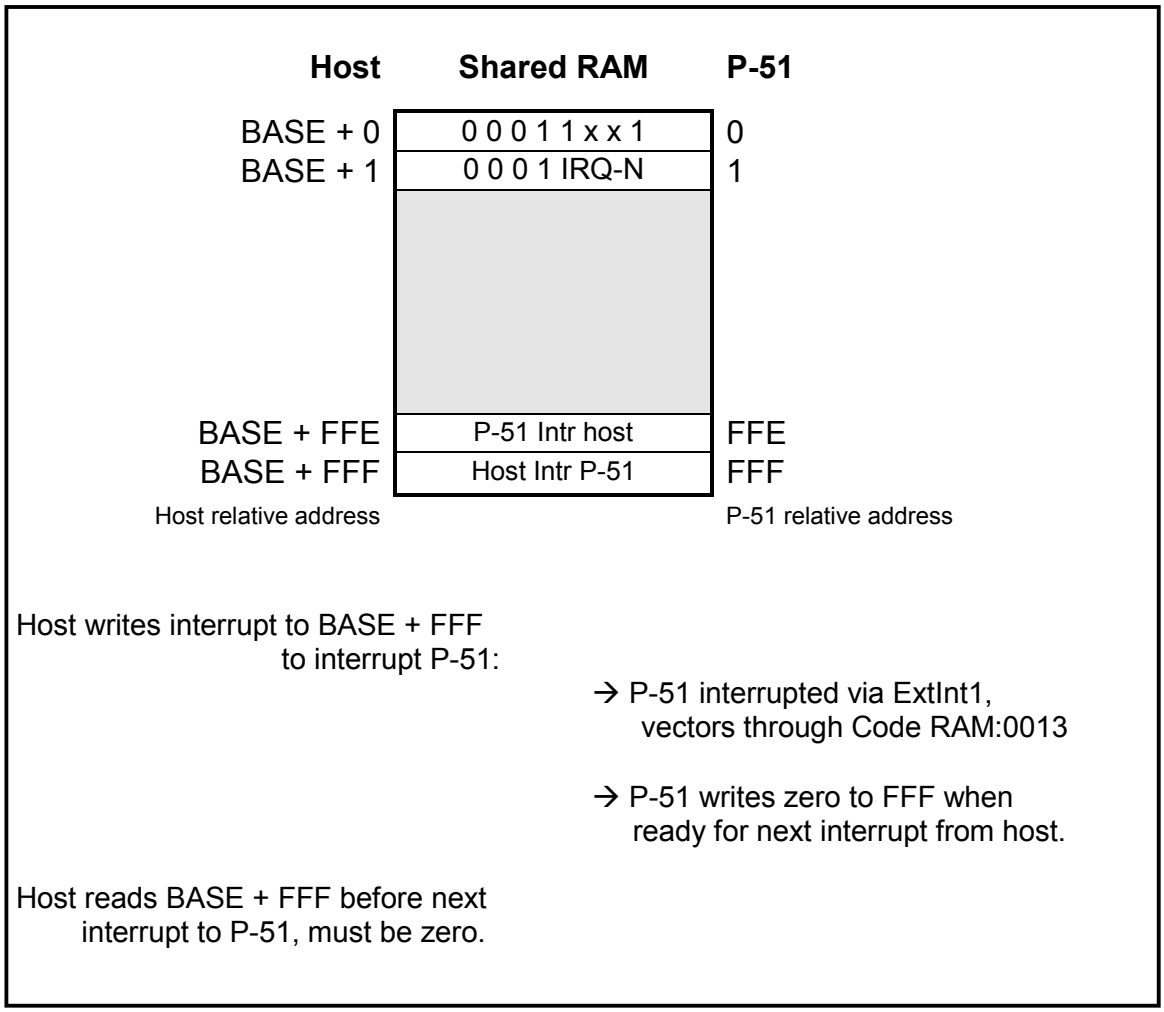
P-51 Protocol to interrupt the host:



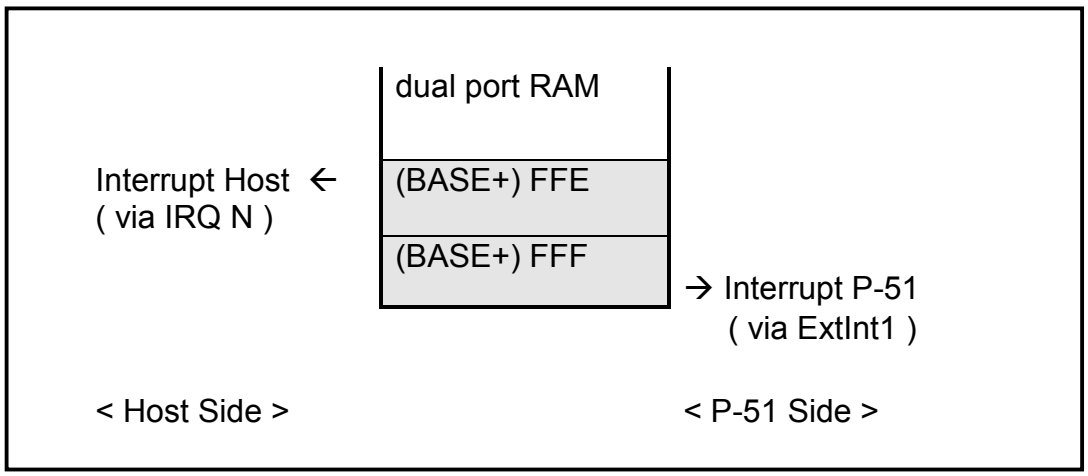
To summarize, the host must setup the IRQ Control Register and the Master Control Register to allow interrupts to the host on a specified IRQ-pin. Once this has been done, the P-51 can interrupt the host by writing any (non-zero) value into address 0xFFE in shared RAM space, causing the selected IRQ-pin to pulse low. The host should respond to this IRQ pulse by reading the value of location (BASE +) 0xFFE and interpreting the value (the interpretation is by a pre-agreed convention between P-51 program and host). When the host has finished and is able and willing to accept more interrupts the host writes zero to (BASE +) 0xFFE to signal READY to the P-51. The P-51 should test location 0xFFE for zero before writing a value to this address. If the value is non-zero, the P-51 should wait for the value to become zero before writing a non-zero interrupt value to 0xFFE.

Host Software Interrupt of P-51:

The above description details how the P-51 software can generate interrupts to the host via an IRQ pin. The converse operation allows the host software to interrupt the P-51 by writing to location (BASE+) 0xFFF. This write will generate an interrupt on P-51 Port 3 Pin 3 (P3.3) which vectors through 8051 External Interrupt 1 location 0x13 (hex) in P-51 Code RAM. The interrupt will cause P3.3 to go low and remain low until the P-51 reads location 0xFFF in shared space (dual port RAM). The act of reading 0xFFF (from the P-51 side) clears the interrupt and P3.3 returns hi. The P-51 can interpret the value read from 0xFFF to decide what action is being requested. Finally the P-51 can write zero to 0xFFF, signifying to the host that the P-51 is ready to accept further interrupts.

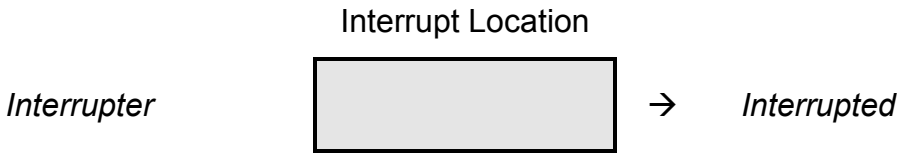


Summary of Software Controlled Interrupts:



The Software Interrupts are symmetrical and are supported by P-51 hardware and extended by software protocols.

The hardware support allows the interrupts to write to a location and thereby generate an interrupt to the "other side". If the other side **writes** to the same location there is no effect except to write a value into memory, however if the other side **reads** the same location, then the effect is to clear the interrupts. When the interrupting side reads the same location, there is no effect:



Hardware Mechanism:

Write: Interrupt	Write: No Effect
Read: No Effect	Read: Clear Interrupt

Software Protocol:

Write: Interrupt Request Value	Write: To Signal Ready (=0)
Read: see if other side is ready	Read: To decode interrupt request value.

P-51 Access to Code RAM

When the P-51 is released from reset (the Control Reg must contain 0001 xxx1) its Program Counter (PC) is set to zero. This causes the P-51 to fetch the first byte from Code RAM and begin executing code. (Normally the first instruction is a "jump over vectors"). During execution, the P-51 fetches all instructions from Code RAM using the Program Counter. The P-51 has only one access mechanism to Code RAM. The 8051 "movc a,@a + Dptr" allows the P-51 to use a value in the accumulator as an index into a table that Dptr points to, and read from the table into the accumulator. There is no other P-51 access to Code RAM.

Host Access to Code RAM:

The host processor can, while the P-51 is held in Reset, access Code RAM at locations BASE + 0x1000 to BASE + 0x2FFF, using normal read and write operations from the EISA-bus side of Code RAM. There is no other host access to Code RAM.

P-51 Access to Dual Port RAM:

The P-51 uses the 8051 "movx" instruction to access its own internal dual port RAM. Those familiar with the 8051 will remember that "movx" is normally used in conjunction with P2 and P0 to access external memory or devices outside the 8051. This is still the case in the P-51, however it is complicated by the existence of the integral dual port RAM. The solution is as follows:

The P-51 I/O pin port 1.6 is used to select between dual port RAM and an external memory or device. When P1.6 is set (as it is by 'RESET') then the P-51 "movx" instructions address external memory (via the "CE-out" pin 73). When P1.6 is cleared, as it is by the "clr P1.6" instruction, then "CE-out" is disabled (i.e., goes to '1') and the internal dual port RAM is enabled. When dual port RAM is enabled, all "movx" operations operate on dual port RAM. These include:

```
movx a, @ RN      ;  read thru P2:(Rn)
movx a, @ Dptr    ;  read thru (Dptr)
movx @ Rn, a      ;  write thru   P2:(Rn)
movx @ Dptr, a    ;  write thru (Dptr)
```

Movx Latch: The "movx" instructions in the P51 are identical to the "movx" instructions in the 8051, with respect to the read and write signals, P3.6 and P3.7, and to address and data on P2 and P0. This is true whether the target of the "movx" is internal or external; that is, whether the P51 is accessing the internal dual port RAM or an external RAM/IO-device.

A consequence of this is that systems using only internal dual port RAM will still see P2:P0 change (and P3.6,P3.7) when "movx" occurs. If these ports are intended to drive external devices, then latches may be required on P2 and P0 to shield the external devices from "movx" induced changes.

Similarly, if external RAM-I/O is used, the external RAM must be disabled during internal "movx" operations. This will normally be done using P1.6 as chip select

P-51 Dual Data Pointers:

The P-51 possesses two 16-bit Data Pointers instead of the single 16-bit Dptr of the standard 8051. These data pointers are accessed 'one at a time' and are selected by setting or clearing a bit in a special function register, as follows.

The Auxiliary Data Pointer control register exists in Special Function Register space at address **0xA2**. The least significant bit, b0, of this register controls which data pointer is in use. The data pointers are identical in behavior, although the second data pointer is only accessible when selected and has no separate SFR location. Their utility lies in the ability to pre-load both data pointers, then switch between them by toggling the selection bit in the special function registers. This precludes the need, with only a single data pointer, to point to a first location, then save the address, point to a second location, etc. An example of the use of two data pointers to move data from a source buffer to a destination buffer is shown below.

```
; CONSTANTS:

            SRC_buffer   =  0x400      ; location of source bytes
            DST_buffer   =  0x500      ; location of destination buffer
            buf_length   =    0x40     ; number of bytes to move

            DP_sel       data 0xA2     ; Special function register

; CODE:
            ;
            mov Dptr, # SRC_buffer     ; setup source pointer
            inc   DP_sel               ; switch data pointers
            mov Dptr, # DST_buffer     ; setup destination ptr
            mov r0, # buf_length

      Copy_Loop:
            ;
            inc     DP_sel             ; switch to source pointer
            movx  a, @ Dptr            ; get source byte
            inc     Dptr               ; next source location
            inc     DP_sel             ; switch to dest. pointer
            movx @ Dptr, a             ; put byte into dest buf.
            inc     Dptr               ; next destination location
            ;
            djnz    r0, Copy_Loop      ; repeat til done…
            ;                          ; fall thru with Dptr -> destination
            …
```

P-51 SQRT function

The P-51 possesses a Square Root function that takes a 16-bit value and returns an 8-bit square root. This function does not exist on any other 8051, so we describe its operation below:

The P-51 **SQRT** operation utilizes special locations in the Special Function Register address space. The 16-bit value is written into two of these locations and the 8-bit square root is read from a third. The square root is computed in one instruction cycle, so that the answer may be read as soon as the value has been loaded. The Special Function Register locations involved are shown below:

Special Function Space

Reg Name	:	Reg Address
	:	
SQRT_lo		0x84
SQRT_hi		0x85
SQRT		0x86
	:	
	:	

The code fragment below illustrates the use the **SQRT** function:

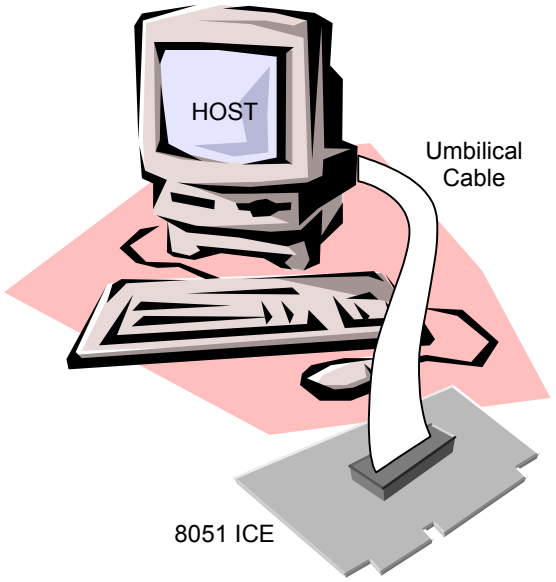
```
SQRT_lo     data 0x84
SQRT_hi     data 0x85
SQRT        data 0x86

mov   a, # 4              ; get hi-byte
mov   SQRT_hi, a          ; write hi-byte
mov   SQRT_lo, # 0        ; write lo-byte
mov   a, SQRT             ; read answer
```

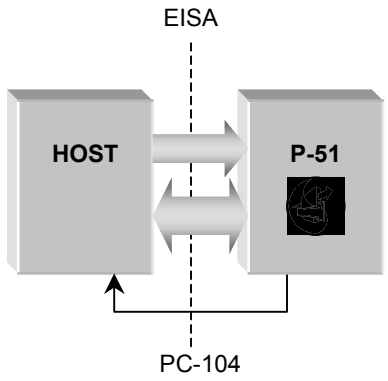
Note that the data can be loaded directly as an immediate (constant) value or can be loaded from the accumulator, allowing variable data to picked up from memory and loaded into the square root circuits. The answer can be read from the SQRT register into the accumulator, then used as appropriate. Any direct moves to and from the SQRT registers can be used.

"Debug" Features of the P-51

The classical 8051 does not support debugging with either breakpoints or single step capability. Instead, special "bond out" versions of the chips are packaged by specialist companies and sold as "ICE", (In-Circuit-Emulation) tools, typically for thousands of dollars. The ICE is controlled by a host connected by an umbilical cord to the 8051 bond-out chip that is plugged into the system.



Classical "ICE" – expensive kludge-based on umbilical extension.



P-51 Debugging – natural extension

P-51 Debug Support

The P-51 offers these features that support debugging 8051 software. None of these features is available on standard 8051s.

- Breakpoint operation
- Single-step operation

Each of these features allows the user, via the host processor, to detect P-51 code execution or data access details that are useful in debugging software. All these features are patented or patent pending.